

(19) 日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開 2002-229778

(P2002-229778A)

(43) 公開日 平成 14 年 8 月 16 日 (2002.8.16)

(51) Int.Cl'	識別記号	F I	データコード(参考)
G 06 F 9/38	3 3 0	G 06 F 9/38	3 3 0 F 5 B 0 1 3
	3 1 0		3 1 0 A 5 B 0 3 3
9/32	3 2 0	9/32	3 2 0 B

審査請求 未請求 請求項の数 4 OL (全 7 頁)

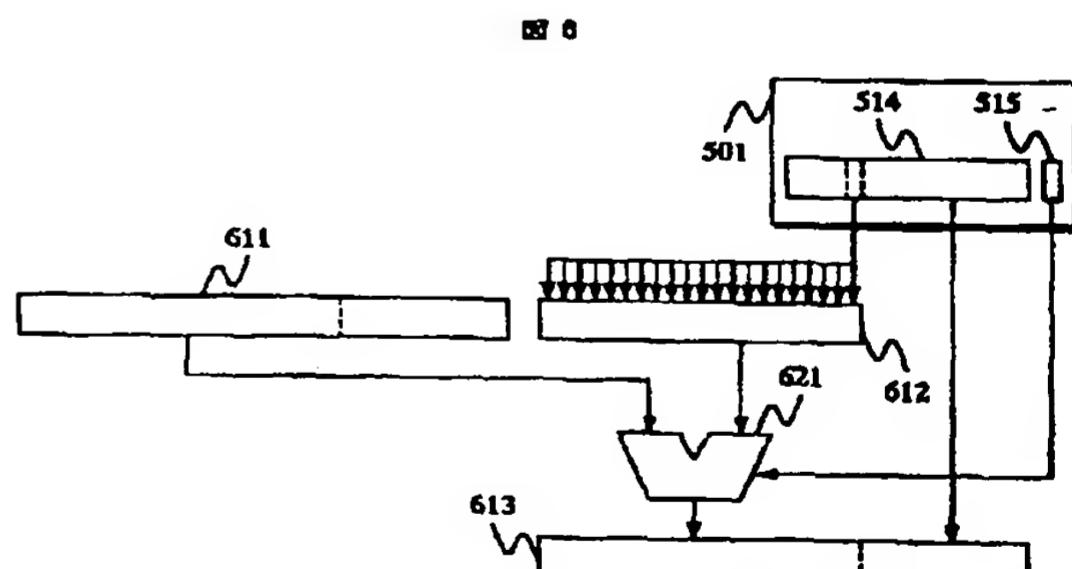
(21) 出願番号	特願 2001-26253 (P2001-26253)	(71) 出願人	000005108 株式会社日立製作所 東京都千代田区神田駿河台四丁目 6 番地
(22) 出願日	平成 13 年 2 月 2 日 (2001.2.2)	(72) 発明者	近藤 雄樹 東京都国分寺市東恋ヶ窪一丁目 280 番地 株式会社日立製作所中央研究所内
		(72) 発明者	西井 修 東京都国分寺市東恋ヶ窪一丁目 280 番地 株式会社日立製作所中央研究所内
		(74) 代理人	100075096 弁理士 作田 康夫 F ターム(参考) 5B013 AA05 BB13 5B033 CA04

(54) 【発明の名称】 高速ディスプレースメント付き PC 相対分岐方式

## (57) 【要約】

【課題】既存のプロセッサの命令動作を変更、もしくは新規命令を追加することなく、ディスプレースメント付き PC 相対分岐命令の動作を高速化する。

【解決手段】命令語をキャッシュもしくはバッファに格納する前にあらかじめ分岐先アドレス計算の下位部分を行い、命令語のディスプレースメント部分及び該キャッシュもしくは該バッファに追加されたビットに計算結果を書き込み、命令実行時は該キャッシュもしくは該バッファに格納された計算結果を利用して分岐先アドレスの下位部分の計算を省略することで分岐先アドレス計算を高速化する。



行においては分岐先アドレス計算の次に命令キャッシュへのアクセスが必要となるが、通常キャッシュの動作は（1）アドレスのうちインデックス部となる部分をキャッシュアレイに入力し対応するタグ部とデータを読み出し、（2）アドレスのタグ部と対応する部分と（1）で出力されたタグ部を比較し、一致する場合キャッシュヒットとしてデータを出力するという2段階に分かれている。すなわちキャッシュへのインデックス部入力は、タグ部の入力よりもタイミング上先に行う必要がある。また、アドレスのうちインデックス部となる部分は、通常はアドレスの下位部分に配置されている。SemiABS分岐では分岐先アドレスの下位部分は加算を行うことなくそのまま出力されるため、インデックス部のビット範囲が加算されないビット範囲に含まれる時、分岐先アドレスの上位部分の計算の完了を待たずに並列にキャッシュアレイのアクセスを開始することができ、分岐命令の実行が全体として高速化される。

#### 【0003】

【発明が解決しようとする課題】上記の従来技術では高速な分岐を実現するために、PC+disp分岐の代わりに新しいSemiABS分岐を導入している。しかし、SemiABS分岐論理方式を採用する新たなプロセッサではSemiABS分岐を採用していない命令によって書かれたプログラムを実行できない、すなわち、既存のPC+disp分岐命令を有するプロセッサ上で動作するオブジェクトコードにはSemiABS分岐命令が含まれていないため、高速化の恩恵を受けることができない。また、SemiABS分岐では分岐先アドレスの一部分を絶対アドレスで指定する必要があるため、コンパイル時に分岐先アドレスの範囲に収まっているかどうかが判定できない、コンパイル後のオブジェクトコードがリロケータブルにならない、という弊害がある。リロケータブルとは、オブジェクトコードの配置アドレスに関わらず、そのオブジェクトコードが実行可能であることを指す。本発明が解決しようとする課題の1つは、以上の問題に鑑みプロセッサの命令セットを新規に用意、または既存の命令セットに命令を追加することなく、PC+disp分岐命令の動作を高速化することである。また、PC+disp分岐命令を削除する必要がないため、上記に示した弊害についても発生しない。

#### 【0004】

【課題を解決するための手段】本発明は命令語をキャッシュもしくはバッファに一時的に格納し、命令の実行時はキャッシュもしくはバッファに一時的に格納されている命令語を読み出して実行するプロセッサに適用するものである。

【0005】本発明では、命令語をキャッシュもしくはバッファに格納する時にPC+dispアドレッシング命令であるかどうかを判定し、PC+dispアドレッシング命令である場合にはキャッシュもしくはバッファ

への格納前にあらかじめPC+dispの計算の下位部分を行い、命令語のdisp部分とキャッシュもしくはバッファに追加されたビットに計算結果を書き込む。該PC+dispアドレッシング命令をキャッシュもしくはバッファから読み出して実行する時にはPC+disp計算の下位部分はあらかじめ計算された結果をそのまま出力し、下位アドレス部分の計算を省略することにより、PC+disp分岐命令のアドレス計算に要する時間を削減する。以上が本発明の手段である。

#### 【0006】

【発明の実施の形態】以下に、本発明の実施例を2例示す。

（実施例1）図5及び図6は本発明を最も単純な形で実現した例である。図5は命令フェッチ部、すなわち主記憶から読み出した命令語を命令キャッシュに格納する部分である。511は主記憶から読み出した命令語であり、命令フォーマットは図2の112と同等である。本実施例のプロセッサではアドレスは32ビットである。512は命令が格納されている主記憶上のアドレスであり、幅は32ビットである。また、アドレス512は命令語511の実行時のPCの値に等しい。

【0007】501は命令キャッシュである。514は命令語511と同じ幅をもっており、命令キャッシュ501内に複数のエントリが存在する。515は514と1対1で対応しており、幅は1ビットである。回路ブロック502は命令語511とアドレス512を入力とし、出力が命令キャッシュ501内の514及び515に格納される。

【0008】回路ブロック502の動作について説明する。命令語511内の201および202は命令キャッシュ501内の514の対応する部分にそのまま格納される。522はデコーダであり、命令語511内の201がPC+disp分岐命令かどうかを判定し、セレクタ523の出力を切り替える。命令語511内の201がPC+disp分岐命令以外の場合、セレクタ523は命令語511内の203をそのまま出力し、命令キャッシュ501内の514の203に対応する部分にそのまま格納される。命令語511内の201がPC+disp分岐命令を示している場合、命令語511内の203とアドレス512の下位11ビット部分を加算器512で加算し、得られた11ビットの結果をセレクタ523が出力する。命令キャッシュ501内の514の203に対応する部分にはこの結果が格納される。また、加算器521の上位へのキャリービット出力は515に格納される。図6はPC+disp分岐命令実行時の分岐先アドレス計算部である。611はPCの値であり、命令キャッシュ501内に格納された514と515及び611の値を用いて、分岐先アドレス613を求める。514内の203に対応する部分はそのまま分岐先アドレス613の下位11ビット部分に出力される。612

た。実施例1に示したプロセッサにおいては、Semi ABS分岐命令と同様に命令キャッシュへのアクセスを早い時刻に開始することによってPC+disp分岐命令の動作を全体的に高速化できるという第二の利点がある。実施例1に示したプロセッサでの第三の利点は、あるPC+disp分岐命令が命令キャッシュにヒットして繰り返し実行される場合には、分岐先アドレスの下位部分の計算は初回実行時、すなわち命令キャッシュに格納される時にのみ行われ、以降の実行時に重複して計算されることを回避できる点である。図9に従来のPC+disp分岐論理方式の場合と実施例1でのPC+disp分岐論理方式の場合との比較を示す。901は従来のPC+disp分岐論理方式の場合であり、911は命令キャッシュ読み出し、および命令キャッシュミスヒットによる主記憶からの命令語読み出し動作、913は911で読み出した命令語をキャッシュに格納する動作、914は分岐先アドレスの下位部分の加算動作、915は分岐先アドレスの上位部分の加算と分岐先アドレス命令の命令キャッシュからの読み出し動作である。また、912は命令キャッシュヒットの場合の読み出し動作である。902は実施例1でのPC+disp分岐論理方式の場合であり、あるPC+disp分岐命令が命令キャッシュにヒットする状態で繰り返し実行される場合、繰り返し回数が多いほど実施例1でのPC+disp分岐論理方式の方が、実行に要する時間が短縮される。

【0015】実施例1に示したプロセッサの第四の利点は、第一、第二及び第三の利点として挙げた効果を、既存のPC+disp型アドレッシングを備えるプロセッサの命令セットに変更を加えることなく、享受できる点である。本発明の実施例2に示したプロセッサでは、実行時にdispの符号ビットおよび命令フェッチ時の加算の最上位ビットからのキャリービットの再計算を行うため実施例1に示したプロセッサの第一の利点に相当する効果は実質的にはほとんど得られないが、第二、第三および第四の利点に関しては同等に効果が得られる。また、実施例2に示したプロセッサでは、主記憶から読み出した命令語を格納するキャッシュもしくはバッファに追加の記憶領域が不要であるという利点がある。

#### 【図面の簡単な説明】

【図1】従来のPC+disp分岐での分岐先アドレス計算法。

【図2】命令語112のフォーマット。

【図3】従来のSemiABS分岐での分岐先アドレス計算法。

【図4】命令語312のフォーマット。

【図5】本発明の第一の実施例における命令語の命令キャッシュ格納時の処理回路。

【図6】本発明の第一の実施例におけるPC+disp分岐命令実行時のアドレス計算回路。

05 【図7】本発明の第二の実施例における命令語の命令キャッシュ格納時の処理回路。

【図8】本発明の第二の実施例におけるPC+disp分岐命令実行時のアドレス計算回路。

10 【図9】従来のPC+disp分岐論理方式の場合と実施例1でのPC+disp分岐論理方式の場合のタイミングチャートでの比較。

【図10】デコーダ822への入力に対するセレクタ823の出力対応図。

#### 【符号の説明】

15 111-PCの値、112-PC+disp分岐命令の命令語、114-PC+disp命令での分岐先アドレス、121-32ビット加算器。

311-PCの値、312-SemiABS分岐命令の命令語、314-SemiABS分岐命令での分岐先アドレス、321-18ビット加算器。

20 501-命令キャッシュ、511-PC+disp分岐命令の命令語、512-読み出した命令の主記憶上のアドレス値、514-命令キャッシュ内のエントリ、515-514に一対一に対応する記憶領域、521-11ビット加算器、522-デコーダ、523-セレクタ。611-PCの値、613-分岐先アドレス、621-21ビット加算器。

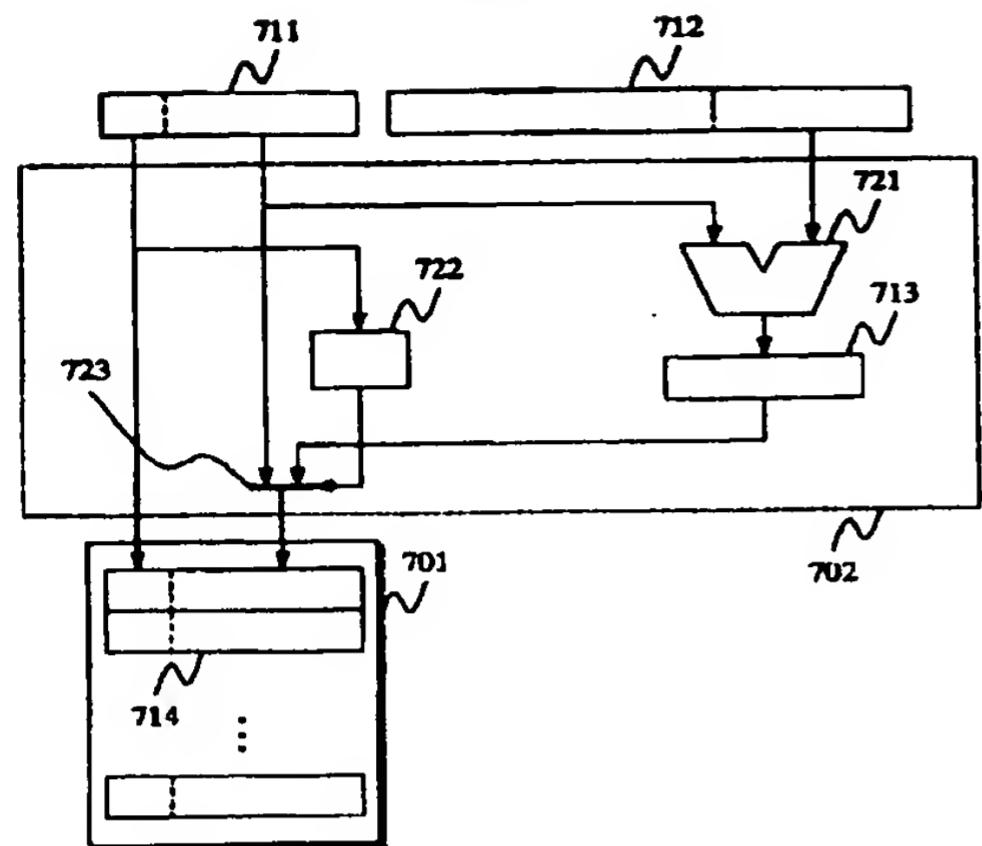
25 701-命令キャッシュ、711-PC+disp分岐命令の命令語、712-読み出した命令の主記憶上のアドレス値、714-命令キャッシュ内のエントリ、721-12ビット加算器、722-デコーダ、723-セレクタ。

30 811-PCの値、812-分岐先アドレス、821-11ビット比較器、822-デコーダ、823-セレクタ、824、825-20ビット加算器。

901-従来のPC+disp分岐論理方式の場合のPC+disp分岐命令動作、902-本発明の第一の実施例でのPC+disp分岐論理方式の場合のPC+disp分岐命令動作、911-命令キャッシュ読み出し及び命令キャッシュミスヒットによる主記憶からの命令語読み出し動作、912-命令キャッシュヒット時の命令語読み出し動作、913-911で読み出した命令語をキャッシュに格納する動作、914-分岐先アドレスの下位部分の加算動作、915-分岐先アドレスの上位部分の加算と分岐先アドレス命令の命令キャッシュからの読み出し動作。

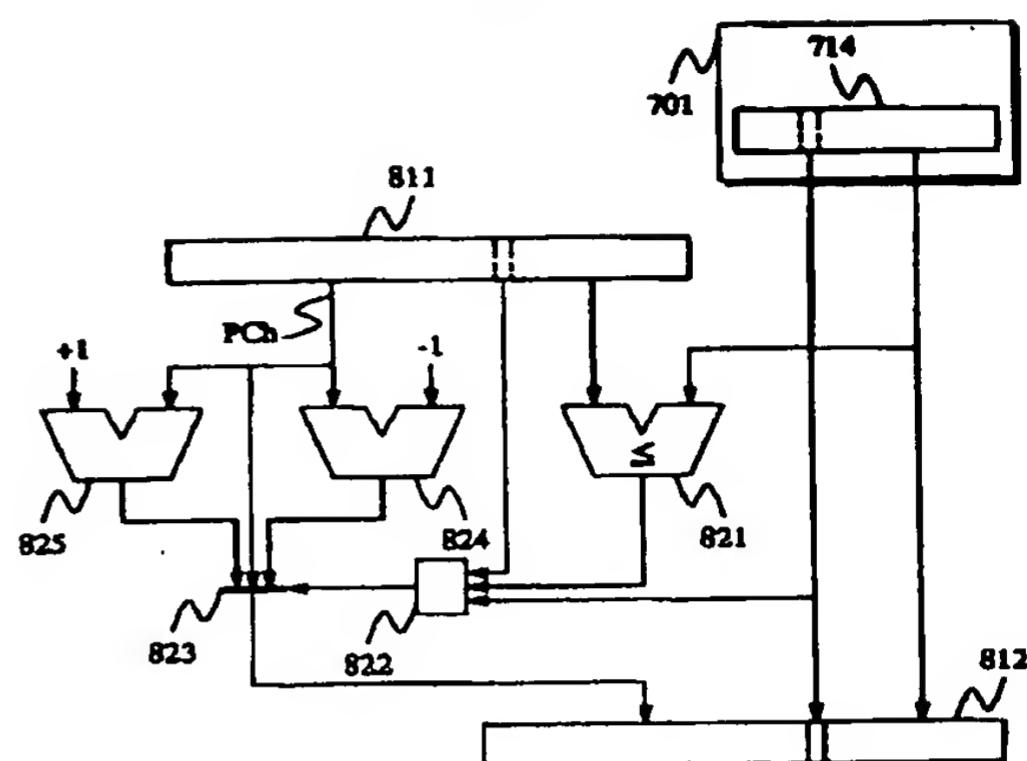
【図 7】

図 7



【図 8】

図 8



【図 10】

図 10

入力	(a)	1	1	1	1	0	0	0	0
	(b)	1	1	0	0	1	1	0	0
	(c)	1	0	1	0	1	0	1	0
出力	セレクタ823 の 出力	PC <sub>a</sub> を 選択	PC <sub>b</sub> を 選択	PC <sub>a+1</sub> を 選択	PC <sub>b</sub> を 選択	PC <sub>a</sub> を 選択	PC <sub>b</sub> を 選択	PC <sub>a</sub> を 選択	PC <sub>b</sub> を 選択